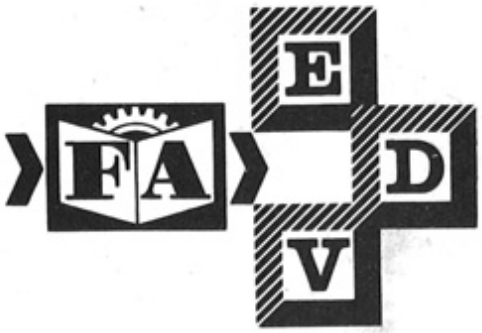
FERNSEHKURS DES DEUTSCHEN FERNSEHFUNKS



Themenkomplex 4/2

Grundlagen der Programmierung

Verlag Die Wirtschaft Berlin

Das Material
entstand unter Verwendung
der Autorenmanuskripte des Fernsehkurses
"Elektronische Datenverarbeitung"

Lektion 18 von Dipl.-Math. Bernhard Labisch, Dresden
Lektion 19 von Dipl.-Math. Klaus Hentschel, Dresden
Lektion 20 von Dr. Karl-Heinz Müller, Dresden
Lektion 21 von Dipl.-Math. Karl-Heinz Naubereit, Dresden

die auch die Überarbeitung
für die Veröffentlichung besorgten

Lektor: Dipl.-Hdl. Renate Eyraud

Redaktionsschluß:
20. August 1969

Verlag Die Wirtschaft, 1055 Berlin, Am Friedrichshain 22
1969 veröffentlicht · Lizenz-Nr. 122 · Druckgenehmigungs-Nr. 195/169/69
Alle Rechte vorbehalten
Schreibsatz: Verlag Die Wirtschaft Berlin
Druck: (140) Neues Deutschland, Berlin
ES 21 A · 0,90 M

Inhaltsverzeichnis

LEKTION 18

Die Programmaufstellung (I)
– Maschinensprache –

1. Einleitung 4
2. Die Arbeitsweise der EDV-Anlage – Grundlage der Programmierung 5
3. Die Grundbefehle einer EDV-Anlage 6
4. Die Programmierung der Lohnberechnung 8
5. Die Merkmale der Programmierung in der Maschinensprache 13

LEKTION 19

Die Programmaufstellung (II)
– Maschinenorientierte Programmiersprache –

1. Einleitung 15
2. Die maschinenorientierte Programmiersprache MOPS 16
3. Die Übersetzung des MOPS-Programms 22
4. Die Merkmale der Programmierung in maschinenorientierter Programmiersprache 25

LEKTION 20

Die Programmaufstellung (III)
– Problemorientierte Programmiersprache –

1. Einleitung 28
2. Die Programmiersprache ALGOL 60 29
3. Die Merkmale der Programmierung in problemorientierter Programmiersprache 34

LEKTION 21

Die Praxis der Programmierung

1. Einleitung 36
2. Die Methoden der praktischen Arbeit der Programmierung 37
3. Die Programmdokumentation 39
4. Die Programmbibliothek 42
5. Die "Benutzergemeinschaft elektronische Datenverarbeitung" 42

LEKTION

18

Die Programmaufstellung (I)
- Maschinensprache -

1. Einleitung

2. Die Arbeitsweise der EDV-Anlage – Grundlage der Programmierung
2. 1. Die Arbeitsweise des Hauptspeichers
2. 2. Die Arbeitsweise des Steuerwerks
2. 3. Die Arbeitsweise des Rechenwerks

3. Die Grundbefehle einer EDV-Anlage
3. 1. Die arithmetischen und die Vergleichsbefehle
3. 2. Die Transportbefehle
3. 3. Die Sprung- und Haltbefehle
3. 4. Die numerische Verschlüsselung der Operationen

4. Die Programmierung der Lohnberechnung
4. 1. Die Aufgabenstellung
4. 2. Der Programmablaufplan
4. 3. Der Speicherbelegungsplan
4. 4. Die Programmierung in der Maschinensprache

5. Die Merkmale der Programmierung in der Maschinensprache

1. Einleitung

Die Lösung einer Aufgabe mit einer elektronischen Datenverarbeitungsanlage setzt ein entsprechendes Programm voraus. Wir wollen uns in dieser Lektion mit der Aufstellung von Programmen – der Programmierung – beschäftigen.

Unter Programmierung verstehen wir das Erarbeiten eines Programms, das heißt einer Folge von Befehlen, die so angeordnet sind, daß der im Programmablaufplan vorgegebene Lösungsalgorithmus von der Datenverarbeitungsanlage abgearbeitet wird.

Da die Befehle in einer der Datenverarbeitungsanlage verständlichen Form vorliegen müssen, verschlüsselt der Programmierer die Anweisungen des Programmablaufplans in Befehle und trägt sie nacheinander in Vordrucke ein. Am Schluß liegt das Programm in Form eines Satzes ausgefüllter Programmvordrucke vor. Daraus entstehen durch Ablochen die Eingabelochkarten (oder das Eingabelochband), über die das Programm in die Datenverarbeitungsanlage vor Beginn der Programmabarbeitung eingespeichert wird.
Von den verschiedenen Möglichkeiten der Formulierung eines Programms wählen wir die Programmierung in der Maschinensprache. Sie steht in engem Zusammenhang mit der Arbeitsweise der EDV-Anlage und bietet den besten Einblick in die Problematik. Andere Möglichkeiten der Programmierung werden wir in den folgenden Lektionen behandeln.

2. Die Arbeitsweise der EDV-Anlage – Grundlage der Programmierung

Die Programmierung in der Maschinensprache setzt eine gründliche Kenntnis der Arbeitsweise der Datenverarbeitungsanlage, für die das Programm aufgestellt wird, voraus. Betrachten wir daher zunächst die Arbeitsweise der Hauptbestandteile einer EDV-Anlage aus der Sicht des Programmierers.

2. 1. Die Arbeitsweise des Hauptspeichers

Der Hauptspeicher einer EDV-Anlage, bestehend aus einer Reihe von Speicherplätzen mit fortlaufenden Adressen, dient sowohl zur Aufnahme des Programms als auch der Daten, mit denen das Programm arbeitet. Bei größeren Datenmengen werden im allgemeinen nur die Daten im Hauptspeicher aufbewahrt, die für den Programmschritt gebraucht werden, während die übrigen Daten auf Abruf in externen Speichern bereitstehen. Die Befehle eines Programms sind in aufeinanderfolgenden Speicherplätzen untergebracht. Sie enthalten die Informationen für die Arbeit des Steuerwerks.

2. 2. Die Arbeitsweise des Steuerwerks

Das Steuerwerk übernimmt auf der Grundlage des gespeicherten Programms die Steuerung der Befehlsabarbeitung. Dabei wiederholt sich ständig folgender Zyklus:

- Befehlsaufruf aus dem Hauptspeicher,
- Befehlsentschlüsselung,
- Ablaufsteuerung der befohlenen Operation (Befehlsabarbeitung).

Ein Befehl enthält im wesentlichen zwei Informationen für das Steuerwerk, und zwar Operation und Adresse.
Durch die Angabe im Operationsteil des Befehls werden durch das Steuerwerk alle Weichen für die Operationsausführung gestellt. Im Adreßteil ist vermerkt, unter welcher Adresse der Operand im Hauptspeicher zu finden ist.

2. 3. Die Arbeitsweise des Rechenwerks

Das Rechenwerk führt die arithmetischen Operationen aus. Der Akkumulator (AC) kann Daten vom Hauptspeicher übernehmen oder an den Hauptspeicher abgeben. Im Akkumulator erscheint jeweils das Ergebnis einer arithmetischen Operation.

Beispiel
Ablauf einer Multiplikation mit zwei Operanden

Beide Operanden (Multiplikand und Multiplikator) stehen zunächst im Hauptspeicher. Während der Multiplikator im Hauptspeicher verbleibt, muß der Multiplikand durch das Programm in den Akkumulator überführt werden. Dann kann im Rechenwerk die Multiplikation durchgeführt werden, wobei in der Steuerung des Ablaufs der Rechenoperation eine Überführung des Multiplikators in das Rechenwerk automatisch erfolgt. Nach Abschluß der Rechenoperation erscheint das Ergebnis – das Produkt – im Akkumulator. Der ursprünglich dort stehende Multiplikand wurde dabei überschrieben. Soll das Ergebnis aufbewahrt werden, muß ein Rücktransport in den Hauptspeicher erfolgen.

3. Die Grundbefehle der EDV-Anlage

Aus der Arbeitsweise einer elektronischen Datenverarbeitungsanlage lassen sich Grundbefehle ableiten, von denen der Programmierer bei der Aufstellung des Programms ausgeht.

3. 1. Die arithmetischen und die Vergleichsbefehle

Arithmetische und Vergleichsbefehle einer EDV-Anlage sind nötig, um arithmetische Rechenoperationen (Addition, Subtraktion, Multiplikation und Division) oder Größenvergleiche von Zahlen durchzuführen. Bei den arithmetischen Operationen wollen wir nicht zwischen Rechnungen mit Festkomma und Gleitkomma unterscheiden, sondern eine der beiden Möglichkeiten als gegeben annehmen. Die Wirkung der einzelnen Operationen zeigt Abbildung 1.
Der Hauptspeicherplatz wird im Adreßteil des Befehls angegeben, das Ergebnis der Rechnung erscheint im Akkumulator.
Durch die Vergleichsoperation „<" oder „>" wird ein arithmetischer Vergleich der Zahl im Akkumulator mit einer Zahl im Hauptspeicher vorgenommen. Das Ergebnis dieser beiden Operationen ist eine der beiden möglichen Aussagen "ja" oder "nein", die, numerisch verschlüsselt, in einer speziellen Baugruppe des Rechenwerks, dem Vergleicher, aufbewahrt wird und dort durch andere Befehle abgefragt werden kann.

Operation	Wirkung
+	< AC > + < HS >
−	< AC > − < HS >
x	< AC > · < HS >
/	< AC > : < HS >
<	< AC > < < HS >
>	< AC > > < HS >

< HS > bedeutet: „Inhalt des Hauptspeicherplatzes"
< AC > bedeutet: „Inhalt des Akkumulators"

Abbildung 1
Arithmetische und Vergleichsbefehle

3. 2. Die Transportbefehle

Transportbefehle bewirken einen Datentransport zwischen Hauptspeicher und Akkumulator oder zwischen Hauptspeicher und Ein- und Ausgabegeräten. Im ersten Fall sprechen wir von internen, im zweiten Fall von externen Transportbefehlen. Wir wollen annehmen, daß bei jedem Transportbefehl immer genau eine Zahl transportiert wird.
Die Wirkung dieser Operationen zeigt Abbildung 2.

Operation	Wirkung
Transport vom Hauptspeicher	< HS > → < AC >
Transport zum Hauptspeicher	< AC > → < HS >
Eingabe in den Hauptspeicher	< E > → < HS >
Ausgabe vom Hauptspeicher	< HS > → < A >

< E > bedeutet: „Inhalt des Eingabegerätes"
< A > bedeutet: „Inhalt des Ausgabegerätes"

Abbildung 2
Transportbefehle

3. 3. Die Sprung- und Haltbefehle

Sprungbefehle realisieren Verzweigungen im Programm. Trifft das Steuerwerk bei der Programmabarbeitung auf einen Sprungbefehl, so verläßt es an dieser Stelle die Befehlsfolge und setzt die Abarbeitung des Programms mit dem Befehl fort, dessen Adresse im Adreßteil des Sprungbefehls (Steueradresse) angegeben ist.
Unbedingte Sprungbefehle werden unmittelbar ausgeführt, bedingte Sprungbefehle nur dann, wenn der vorhergehende Vergleichsbefehl eine zutreffende Aussage ergab.
Die Operation "Halt" veranlaßt das Steuerwerk, die Abarbeitung des Programms zu stoppen. Sie erscheint zum Beispiel am Ende eines Programms.

Abbildung 3 zeigt die Wirkung der einzelnen Operationen.

Operation	Wirkung
Nein-Sprung	bedingter Sprung bei Vergleichsaussage „nein"
Sprung	unbedingter Sprung
Halt	Programmabarbeitung stoppt

Abbildung 3
Sprung- und Haltbefehle

3. 4. Die numerische Verschlüsselung der Operationen

Die in einem Befehl enthaltenen Informationen werden dem Steuerwerk vom Programmierer numerisch vorgegeben. Bei der Adresse im Adreßteil eines

Befehls handelt es sich um Speicherplatznummern. Aber auch die Operation im Operationsteil des Befehls muß numerisch verschlüsselt werden. Für die einzelnen Operationen sind somit Schlüsselzahlen festzulegen, die der Programmierer bei der Notierung der Befehle verwendet. In den Abbildungen 4 bis 6 ist der numerische Code für die Befehle angegeben, die wir für die Programmierung des Beispiels "Berechnung des Bruttolohns und des SV-Beitrags" benötigen.

Abbildung 4
Code der arithmetischen und Vergleichsbefehle

20	< AC >	+	< MS >
30	< AC >	·	< MS >
08	< AC >	>	< MS >

Abbildung 5
Code der Transportbefehle

10	< MS >	→	< AC >
11	< AC >	→	< MS >
19	< E >	→	< MS >
18	< MS >	→	< A >

Abbildung 6
Code der Sprung- und Haltbefehle

00	bedingter Sprung (Nein-Sprung)
01	unbedingter Sprung
02	Halt

4. Die Programmierung der Lohnberechnung

4. 1. Die Aufgabenstellung

Wir wollen bei der Aufstellung des Programms auf das Beispiel "Berechnung des Bruttolohns und des Beitrags für die Sozialversicherung eines Beschäftigten" zurückgreifen.
Für die Berechnung gilt:

B = AS x SL	
SV = 60,00 M	wenn B > 600,00
SV = B x 0,1 M	wenn B ≤ 600,00

SL = Stundenlohn
AS = geleistete Stunden im Monat
B = gesuchter Bruttolohn
SV = gesuchter SV-Beitrag

4. 2. Der Programmablaufplan

Der Programmablaufplan für diese Aufgabenstellung ist Ihnen aus Lektion 16 bekannt. Bei einem Vergleich werden Sie jedoch feststellen, daß wir den Lösungsweg etwas verändert haben, um Ihnen eine andere Variante, die zum gleichen Ergebnis führt, zu zeigen.

Die einzelnen Sinnbilder haben wir von 1 bis 6 numeriert.

Sinnbild 1
besagt, daß die Zahlen AS und SL einzugeben sind.

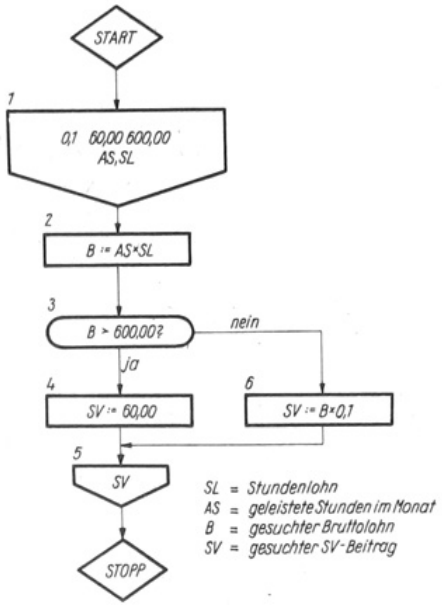


Abbildung 7
Programmablaufplan

Sinnbild 2
fordert die Berechnung von B.

Sinnbild 3
verlangt eine Entscheidung, ob B > 600,00 ist.

Sinnbild 4 oder 6
realisiert die Berechnung von SV in Abhängigkeit von der Aussage in Sinnbild 3.

Sinnbild 5
bedeutet, daß SV auszugeben ist.

Für die Programmierung benötigen wir folgende Grundbefehle:

Sinnbild 1
Transportbefehle (Eingabe)

Sinnbild 2
arithmetische und Transportbefehle

Sinnbild 3
Vergleichs- und Sprungbefehle

Sinnbild 4 und 6
arithmetische und Transportbefehle

Sinnbild 5
Transportbefehle (Ausgabe)

4. 3. Der Speicherbelegungsplan

Bevor wir die einzelnen Befehle des Programms niederschreiben, müssen wir festlegen, unter welchen Adressen die Operanden im Hauptspeicher zu finden sind. Wir stellen einen Speicherbelegungsplan auf.

Zunächst verschaffen wir uns mit Hilfe des Programmablaufplans einen Überblick über die Daten des Programms.

Eingabedaten:	AS, SL		
Konstanten:	0,1	60,00	600,00
Rechendaten:	B, SV		
Ausgabedaten:	SV		

Für diese Daten muß im Hauptspeicher der Datenverarbeitungsanlage Speicherplatz vorgesehen werden. Da der Speicher nach Zeichen adressiert ist und jede Ziffer speicherplatzmäßig einem Zeichen entspricht, setzt sich der Zeichenbedarf für die Daten des Programms wie folgt zusammen:

0,1	2 Zeichen
60,00	4 Zeichen
600,00	5 Zeichen
AS	3 Zeichen
SL	3 Zeichen
B	6 Zeichen
SV	4 Zeichen

Für das Komma wird keine besondere Speicherstelle benötigt.

Wir gehen davon aus, daß die Anzahl der Arbeitsstunden je Monat maximal dreistellig ist, ebenso der Stundenlohn SL (eine Stelle vor und zwei nach dem Komma). Ähnliche Überlegungen führen zu der angegebenen Zeichenanzahl für B und SV.
Nehmen wir an, daß bis zur Adresse 999 der Speicher mit anderen Daten belegt ist, die mit der Aufgabenstellung "Lohnberechnung" nichts zu tun haben. Dann hat der erste freie Speicherplatz die Adresse 1000, hier können wir mit der Abspeicherung beginnen. Wir erhalten folgenden Speicherbelegungsplan:

Daten	Adresse
0,1	1000
60,00	1002
600,00	1006
AS	1011
SL	1014
B	1017
SV	1023

4.4. Die Programmierung in der Maschinensprache

Schlagen Sie bitte den Anhang dieser Lektion auf. Dort finden Sie neben dem Speicherbelegungsplan, dem Programmablaufplan und einer Zusammenstellung der benötigten Operationen einen Vordruck, in den Sie die Befehle des Programms selbst eintragen können.

Wir nehmen an, daß die Zahlen 0,1, 60,00, 600,00, AS und SL in dieser Reihenfolge in das Eingabegerät einlaufen und durch einen Eingabebefehl jeweils eine dieser Zahlen dem Eingabegerät entnommen wird.

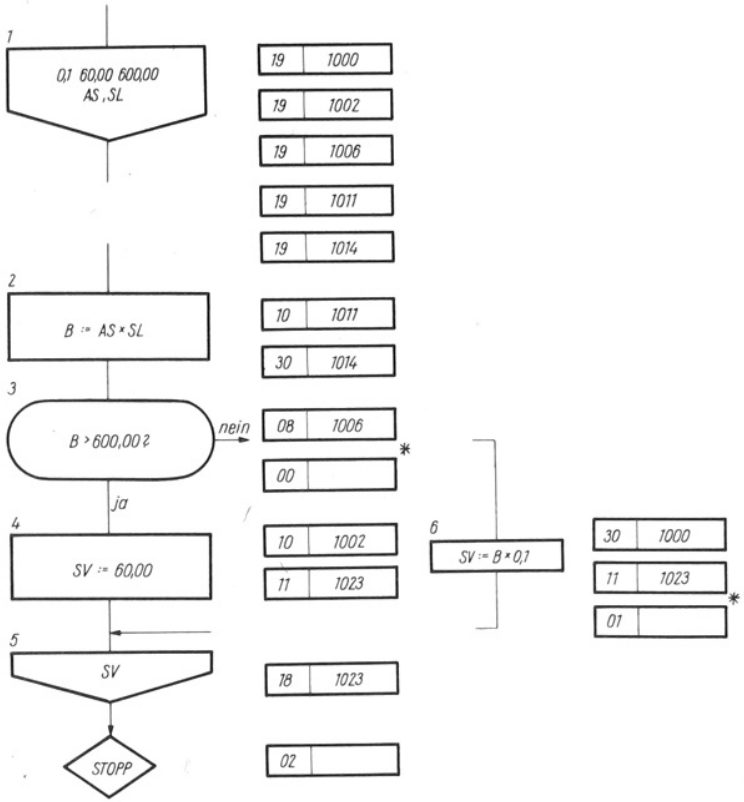


Abbildung 8
Programmentwurf

Beachten Sie, daß
- der Inhalt eines Speicherplatzes oder des Akkumulators nur durch einen Einspeichervorgang, nicht aber durch einen Entnahmevorgang verändert wird;
- durch eine Vergleichsoperation keiner der beiden beteiligten Operanden verändert wird.

Um das Entstehen des Programms zu verdeutlichen, stellen wir die Sinnbilder des Programmablaufplans und die ihnen entsprechenden Befehle nebeneinander (vgl. Abbildung 8).
In den beiden, in Abbildung 8 mit einem Stern gekennzeichneten Sprungbefehlen sind die Steueradressen noch einzutragen. Dazu müssen wir festlegen, in welchen Speicherplätzen die Befehle unseres Programms stehen sollen. Nehmen wir an, das Programm sei anschließend an die Daten zu speichern und jeder Befehl bestehe aus 6 Zeichen – belege also 6 Speicherplätze –, so erhalten wir die in Abbildung 9 gezeigte Darstellung.
Damit können wir die Steueradressen der beiden Sprungbefehle 1075 und 1117, die noch offen waren, zu 1105 und 1093 bestimmen und im Adreßteil der beiden Befehle eintragen. Außerdem können wir jetzt feststellen, daß der Speicherplatz für die Größe B vom Programm gar nicht angesprochen wird. Die Adresse 1017 tritt nicht auf. Das wird erreicht, indem die errechnete Größe B sofort im Akkumulator weiterverarbeitet wird, also gar nicht gespeichert zu werden braucht.

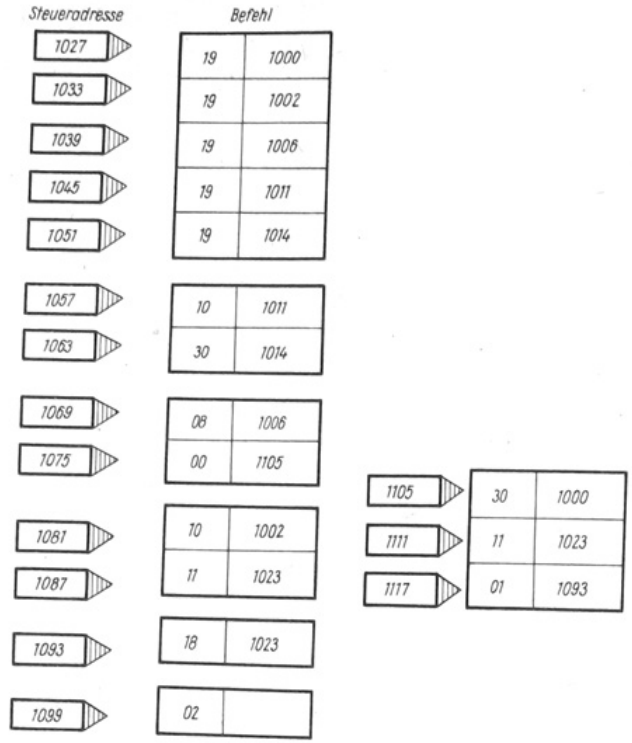


Abbildung 9
Festlegung der
Steueradressen

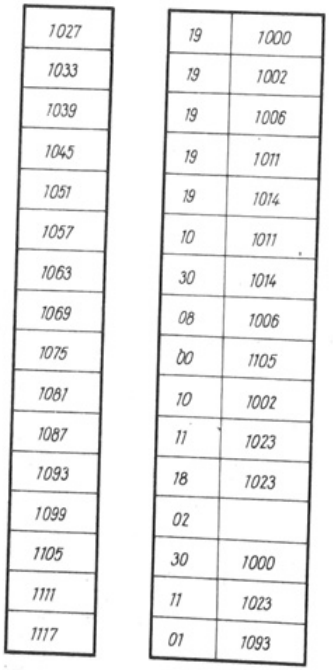


Abbildung 10
Programm

Das Programm erhält seine endgültige Gestalt, wenn wir die Befehle ohne Zwischenräume in der Reihenfolge ihrer Einspeicherung, das heißt ihrer Steueradressen, zusammenstellen.

LEKTION 18

5. Die Merkmale der Programmierung in der Maschinensprache

Bei der Anfertigung des Programms haben wir uns bestimmter Grundbefehle bedient. Sie sind dadurch charakterisiert, daß ihre Operationen in der Anlage "verdrahtet" sind, das heißt, vom Steuerwerk direkt verstanden werden.

Die Menge der möglichen Grundbefehle einer Datenverarbeitungsanlage bezeichnet man als Befehlsliste der Anlage.

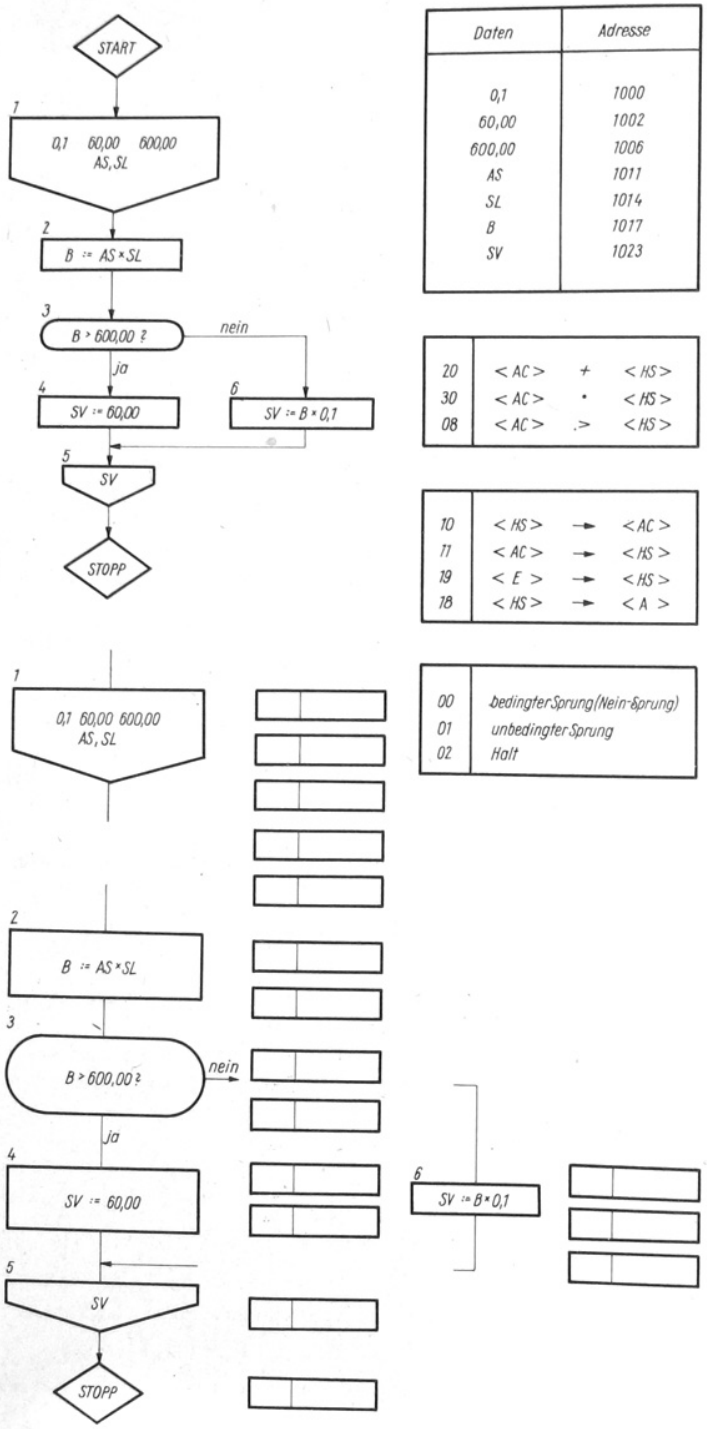
Bei der Programmierung in der Maschinensprache verwendet man die Befehle der Befehlsliste der Anlage. Obwohl die einzelnen Befehlsgruppen, die wir im Abschnitt 3 erläuterten, im Prinzip bei jeder Datenverarbeitungsanlage vorkommen, variiert der Umfang der Befehlsliste von Anlage zu Anlage. Die Befehlsliste des Kleinrechners SER umfaßt etwa 15 Grundbefehle, die des Robotron 300 mehr als 100.
Für die Programmierung in einer Maschinensprache sind folgende Merkmale charakteristisch:

- Der Rechengang einer Aufgabe muß so weit zerlegt werden, daß die entstehenden Operationen durch Befehle aus der Befehlsliste der betreffenden Anlage beschrieben werden können.
- Das in einer Maschinensprache geschriebene Programm besteht aus einer Folge von Befehlen entsprechend der Aufgabenstellung. Jeder Befehl setzt sich aus dem Operationsteil und dem Adreßteil zusammen.
- Der Programmierer muß einen Speicherbelegungsplan aufstellen, in dem festgelegt wird, wo Daten und Programmbefehle gespeichert werden sollen.
- Der Operationsteil der Befehle enthält die Operation in numerisch verschlüsselter Form.
- Der Adreßteil der Befehle nennt die Adresse, daß heißt die Speicherplatznummer, auf die sich die betreffende Operation bezieht.
- Im Adreßteil von Sprungbefehlen wird die Steueradresse, die Speicherplatzadresse des anzusteuernden Befehls, zahlenmäßig angegeben.

LEKTION

18

Anhang



Daten	Adresse
0,1	1000
60,00	1002
600,00	1006
AS	1011
SL	1014
B	1017
SV	1023

20	< AC >	+	< HS >
30	< AC >	*	< HS >
08	< AC >	.>	< HS >

10	< HS >	→	< AC >
11	< AC >	→	< HS >
19	< E >	→	< HS >
18	< HS >	→	< A >

00	bedingter Sprung (Nein-Sprung)
01	unbedingter Sprung
02	Halt

LEKTION

19

Die Programmaufstellung (II)

- Maschinenorientierte Programmiersprache -

1. Einleitung

2. Die maschinenorientierte Programmiersprache MOPS

2. 1. Die symbolischen Operationen
2. 2. Die symbolischen Adressen
2. 3. Die Aufstellung des Programms in MOPS
2. 3. 1. Der Verarbeitungsteil des Programms
2. 3. 2. Der Vereinbarungsteil des Programms
2. 3. 3. Das vollständige MOPS-Programm

3. Die Übersetzung des MOPS-Programms

3. 1. Die Eingabe des Ursprungsprogramms
3. 2. Der erste Übersetzungslauf
3. 3. Der zweite Übersetzungslauf
3. 4. Die Ausgabe des Objektprogramms

4. Die Merkmale der Programmierung in maschinenorientierter Programmiersprache

1. Einleitung

Die Aufstellung eines Programms, seine Formulierung, geschieht mit Hilfe von Programmiersprachen.

Unter einer Programmiersprache versteht man eine eindeutige Sprache – eine definierte Gruppe von Symbolen und Festlegungen –, die dazu dient, Programme auszudrücken.

In Lektion 18 beschäftigten wir uns mit der Programmierung in der Maschinensprache.

Als Maschinensprache bezeichnet man eine Programmiersprache, die nur Maschinenbefehle enthält.

LEKTION 19

In der Maschinensprache aufgestellte Programme sind maschinenabhängig. Sie können von der Datenverarbeitungsanlage direkt verarbeitet werden. Für die Anfertigung eines derartigen Programms genügt es daher nicht, die Maschinensprache irgendeiner elektronischen Datenverarbeitungsanlage zu kennen, der Programmierer muß die Maschinensprache der Anlage, in der das Programm ablaufen soll, genau beherrschen.
Ein in der Maschinensprache aufgestelltes Programm erfordert viel Programmierzeit, da der Programmierer das zu bearbeitende Problem soweit zerlegen muß, daß Operationen entstehen, die durch Befehle aus der Befehlsliste der Datenverarbeitungsanlage beschrieben werden können. Nachteilig wirkt sich aus, daß die Anwendung der Maschinensprache mit einer direkten Adressierung (absolute Festlegung der Speicherplätze für Befehle) und der zusätzlichen Aufstellung eines Speicherbelegungsplans verbunden ist.

Die Programmierung in der Maschinensprache ist somit sehr aufwändig und fehleranfällig. Es ist deshalb notwendig, nach Mitteln und Wegen der Vereinfachung der Programmierung zu suchen, zumal die Datenverarbeitungsanlagen immer größer und komplizierter und die zu lösenden Aufgaben immer umfangreicher und komplexer werden. Alle Vereinfachungen der Programmierung müssen darauf gerichtet sein, den Programmierer für schöpferische Arbeiten frei zu machen und alle routinemäßigen Arbeiten der elektronischen Datenverarbeitungsanlage zu übertragen.
Eine Vereinfachung der Programmierung bieten maschinenorientierte Programmiersprachen.

I Maschinenorientierte Programmiersprachen sind in ihrem Aufbau keiner speziellen Maschine zugeordnet, sie enthalten die bei den meisten Datenverarbeitungsanlagen vorkommenden Befehlsarten in allgemeiner Form.

In dieser Lektion wollen wir uns mit der maschinenorientierten Programmiersprache MOPS für unsere elektronische Datenverarbeitungsanlage Robotron 300 beschäftigen.
Wir greifen dabei wieder auf das Beispiel "Berechnung des Bruttolohns und des SV-Beitrags" zurück und verwenden die Bezeichnungen und Schreibweisen, die uns schon aus den letzten Lektionen bekannt sind.

2. Die maschinenorientierte Programmiersprache MOPS

Wie bei der Programmierung in der Maschinensprache müssen wir auch in MOPS das zu lösende Problem, ausgehend vom Programmablaufplan, in solche Schritte zerlegen, die die EDV-Anlage ausführen kann. Wir sprechen auch hier von Befehlen, bestehend aus Operationsteil und Adressenteil.

2. 1. Die symbolischen Operationen

In der Maschinensprache mußten wir für die möglichen Operationen numerische Verschlüsselungen angeben.
Wir schrieben zum Beispiel für den Vergleich einer im Akkumulator befindlichen Zahl mit einer im Hauptspeicher stehenden Zahl die Verschlüsselung 08 in den Operationsteil.

Bei der Programmierung in MOPS ist diese Verschlüsselung nicht notwendig. Man schreibt in den Operationsteil stattdessen eine Abkürzung für die auszuführende Operation, für den Vergleichsbefehl die Buchstaben VG.

Für die Operationen, deren numerische Verschlüsselungen der Maschinensprache in Lektion 18 in den Abbildungen 4 bis 6 angegeben sind, legen wir in MOPS folgende Abkürzungen fest:

Befehl	*Symbol*
< AC > + < HS >	AD
< AC > · < HS >	M
< AC > > < HS >	VG

Abbildung 1
Symbolische Schreibweise einiger arithmetischer und Vergleichsbefehle (MOPS)

Befehl	*Symbol*
< HS > → < AC >	TV
< AC > → < HS >	TN
< E > → < HS >	E
< HS > → < A >	A

Abbildung 2
Symbolische Schreibweise einiger Transportbefehle (MOPS)

Befehl	*Symbol*
bedingter Sprung (Nein-Sprung)	SN
unbedingter Sprung	S
Halt	H

Abbildung 3
Symbolische Schreibweise einiger Sprung- und Haltbefehle (MOPS)

Wir erkennen, daß die Buchstabenkombinationen für die einzelnen Operationen direkt aus deren Bezeichnung abgeleitet sind. Dadurch prägen sie sich wesentlich leichter ein als die numerischen Verschlüsselungen der Maschinensprache. Maximal können drei Buchstaben für eine Operation verwendet werden.

2. 2. Die symbolischen Adressen

In der Maschinensprache ist im Adressenteil des Befehls die Nummer des Speicherplatzes anzugeben, auf dem die zu verarbeitende Größe steht. Der Programmierer entnimmt sie aus dem vorher aufgestellten Speicherbelegungsplan. An die Stelle der Speicherplatznummer des Operanden tritt bei der maschinenorientierten Sprache das "Merkmal" des Operanden.

I Unter einem Merkmal wollen wir charakteristische Abkürzungen oder Zeichenfolgen verstehen, die den Operanden kennzeichnen.

Für jeden Operanden ist ein Merkmal festzulegen. Die Wahl der Merkmale ist dem Programmierer überlassen. Er wird sie möglichst so festlegen, daß aus dem Merkmal auf Inhalt oder Bedeutung der Größe geschlossen werden kann. Wenig sinnvoll wäre es zum Beispiel, die Konstante 600,00 M mit dem Merkmal DREI zu kennzeichnen.

Nehmen wir an, wir haben für die Größe 600,00 M aus dem Beispiel "Berechnung des Bruttolohns und des SV-Beitrages" das Merkmal SECHS vereinbart. Wollen wir, wie im Beispiel gefordert, den im Akkumulator stehenden Bruttolohn mit 600,00 M vergleichen, so lautet der Befehl in der maschinenorientierten Sprache

VG SECHS.

Wir geben also im Adressenteil des Befehls das Merkmal des Operanden an, ohne daß wir seine Speicherplatznummer kennen.
Ein Merkmal kann maximal sieben alphanumerische Zeichen (eine Kombination aus Ziffern und Zeichen) umfassen.
Die Merkmale beim Robotron 300 können aus 1-7 Buchstaben oder Ziffern bestehen, es muß jedoch wenigstens ein Buchstabe vorkommen. Reine Zahlen sind als Merkmale nicht zugelassen.
Verwenden wir nun im Beispiel "Berechnung des Bruttolohns und des SV-Beitrages" für die Operanden, denen wir in Lektion 18 im Speicherbelegungsplan bestimmte Adressen zugeordnet hatten, die in Abbildung 4 angegebenen Merkmale.

Größe	*Merkmal*
Konstante 0,1	*1ZEHNT*
Konstante 60,00	*SECHZIG*
Konstante 600,00	*SECHS*
Anzahl der Arbeitsstunden	*AS*
Stundenlohn	*SL*
Bruttolohn	*B*
SV-Beitrag	*SV*

Abbildung 4
Merkmale der Operanden

Dabei haben wir für die letzten vier Größen als Merkmale die Abkürzungen eingesetzt, die auch in der Aufgabenstellung zur Bezeichnung dieser Größen dienten.
Diese Möglichkeit nutzen wir, um eine gute Verbindung zwischen Aufgabenstellung und Programm herzustellen.
Wie mit dem Speicherbelegungsplan bei der Maschinensprache, schaffen wir uns mit der Festlegung der Merkmale bei der maschinenorientierten Sprache die Grundlage für die Programmaufstellung.

2. 3. Die Aufstellung des Programms in MOPS

2. 3. 1. Der Verarbeitungsteil des Programms

Veranschaulichen wir uns die Programmaufstellung am uns bekannten Beispiel "Berechnung des Bruttolohns und des SV-Beitrages".
Die Grundlage auch der maschinenorientierten Programmierung bildet der Programmablaufplan, den wir in den Lektionen 17 und 18 ausführlich behandelten (vgl. Abbildung 7 der Lektion 18).
Wenn Sie das Programm selbständig aufstellen wollen, schlagen Sie bitte den Anhang zu dieser Lektion auf. Sie finden dort den Programmablaufplan und die Übersichten über die Symbole für die Operationen sowie die Merkmale für die Operanden, die Sie für die Programmaufstellung benötigen.

Die Befehle sind in das Schema einzutragen. Sie brauchen zunächst nur die beiden Felder OPERATION und ADRESSE, in denen Sie angeben, welcher Befehl mit welcher Größe auszuführen ist.
Übergehen Sie die Eingabebefehle für die Konstanten 01, 60,00 und 600,00 zunächst.
Bei den Sprungbefehlen lassen Sie die Adresse erst einmal weg.
Die richtige Lösung lautet:

MERKMAL	OPERATION	ADRESSE
	E	AS
	E	SL
	TV	AS
	M	SL
	VG	SECHS
	SN	
	TV	SECHZIG
	TN	SV
	A	SV
	H	
	M	1ZEHNT
	TN	SV
	S	

Tragen Sie nun die fehlenden Angaben ein!
In den beiden Sprungbefehlen SN und S fehlen die Adressen.
SN ist die Operationsbezeichnung für den bedingten Sprungbefehl. Er wird nur ausgeführt, wenn B nicht größer als SECHS, der Bruttolohn also nicht größer als 600,00 M ist.
In der Maschinensprache war in den Adressenteil von Sprungbefehlen die Steueradresse des nächsten abzuarbeitenden Befehls einzutragen. Das wäre in unserem Beispiel die des Befehls M 1ZEHNT. Da Sie diese Steueradresse nicht kennen, geben Sie dem Befehl ein Merkmal und schreiben es vor den Befehl.
Versehen Sie diesen Befehl mit dem Merkmal PROZENT, da hier der 10%ige SV-Beitrag berechnet wird. Das Merkmal PROZENT schreiben Sie in den Adressenteil des Sprungbefehls SN. Damit drücken Sie aus, daß bei erfüllter Sprungbedingung der so markierte Befehl als nächster auszuführen ist.
Ebenso verfahren Sie bei dem unbedingten Sprungbefehl S. Nach diesem Sprungbefehl soll der Ausgabebefehl für den SV-Beitrag ausgeführt werden. Diesem geben Sie wieder ein Merkmal, das Sie links vor den Befehl und in den Adressenteil des unbedingten Sprungbefehls schreiben. Der Bedeutung entsprechend wählen Sie das Merkmal AUSGABE.
Da in einem Programm, das in maschinenorientierter Programmiersprache geschrieben ist, auch der erste Befehl des Programms mit einem Merkmal versehen sein sollte, entscheiden Sie sich für das Merkmal START.
Vervollständigt sieht die Darstellung in der maschinenorientierten Sprache wie folgt aus:

LEKTION

19

MERKMAL	OPERATION	ADRESSE
START	E	AS
	E	SL
	TV	AS
	M	SL
	VG	SECHS
	SN	PROZENT
	TV	SECHZIG
	TN	SV
AUSGABE	A	SV
	H	
PROZENT	M	1ZEHNT
	TN	SV
	S	AUSGABE

Abbildung 5
Verarbeitungsteil des Programms

Diesen Teil des Programms bezeichnen wir als Verarbeitungsteil.

2. 3. 2. Der Vereinbarungsteil des Programms

Wie wir aus den Lektionen über Aufbau und Arbeitsweise einer EDV-Anlage wissen, müssen die so geschriebenen Befehle in die maschineninterne Darstellung umgewandelt werden, bevor das Programm abgearbeitet werden kann. Diese Umwandlung wird durch ein spezielles Übersetzungsprogramm in der EDV-Anlage selbst durchgeführt.
Notwendig für die Umwandlung ist, daß wir dem Übersetzungsprogramm mitteilen, was wir unter den verwendeten Merkmalen verstehen. Das geschieht im Vereinbarungsteil des Programms.
Der Vereinbarungsteil besteht wie der Verarbeitungsteil aus Befehlen, und zwar aus Definitionsbefehlen. Mit ihrer Hilfe vereinbaren wir die für den Verarbeitungsteil des Programms erforderlichen Operanden.
Wir unterscheiden Definitionsbefehle für Konstanten und für Variablen.

Der Definitionsbefehl "Definiere Konstante (DK)"

In den Operationsteil dieses Befehls schreiben wir die Buchstaben DK. Im Adressenteil geben wir an, welche Konstante wir mit diesem Befehl festlegen.
Im Beispiel "Berechnung des Bruttolohns und des SV-Beitrages" brauchen wir die Konstante 600,00 M. Der Definitionsbefehl dafür lautet:

DK 600,00.

Um mit dieser Konstanten arbeiten zu können, schreiben wir links vor den Befehl das Merkmal SECHS, auf das wir uns geeinigt hatten. Somit lautet der volle Definitionsbefehl zur Festlegung dieser Konstanten:

SECHS DK 600,00.

In gleicher Weise definieren wir die anderen Konstanten.

MERKMAL	OPERATION	ADRESSE
1ZEHNT	DK	0,1
SECHZIG	DK	60,00
SECHS	DK	600,00

Abbildung 6
Konstantendefinitionen

Der Definitionsbefehl "Definiere Bereich (DB)"

Diesen Befehl verwenden wir für die Festlegung der Variablen.
Die Abkürzung DB wird in den Operationsteil eingetragen. Im Adressenteil wird die erforderliche Bereichsgröße in Zeichen angegeben.
Zum Beispiel benötigt man für die Anzahl der Stunden im Monat einen dreistelligen und für den SV-Beitrag einen vierstelligen Bereich (zwei Stellen vor, zwei Stellen nach dem Komma). Auch hier schreibt man das Merkmal des Bereichs vor den Definitionsbefehl.

MERKMAL	OPERATION	ADRESSE
AS	DB	3
SL	DB	3
B	DB	6
SV	DB	4

Abbildung 7
Bereichsdefinitionen

Der vollständige Vereinbarungsteil des gewählten Beispiels sieht wie folgt aus:

MERKMAL	OPERATION	ADRESSE
1ZEHNT	DK	0,1
SECHZIG	DK	60,00
SECHS	DK	600,00
AS	DB	3
SL	DB	3
B	DB	6
SV	DB	4

Abbildung 8
Vereinbarungsteil des Programms

2. 3. 3. Das vollständige MOPS-Programm

Vereinbarungsteil und Verarbeitungsteil bilden zusammen das MOPS-Programm, das in der Abbildung 9 vollständig enthalten ist.

Jedes MOPS-Programm wird mit dem Befehl END abgeschlossen.

In seinem Adressenteil ist das Merkmal des ersten Befehls des Verarbeitungsteils anzugeben. In unserem Beispiel war das das Merkmal START.

MERKMAL	OPERATION	ADRESSE
1ZEHNT	DK	0,1
SECHZIG	DK	60,00
SECHS	DK	600,00
AS	DB	3
SL	DB	3
B	DB	6
SV	DB	4
START	E	AS
	E	SL
	TV	AS
	M	SL
	VG	SECHS
	SN	PROZENT
	TV	SECHZIG
	TN	SV
AUSGABE	A	SV
	H	
PROZENT	M	1ZEHNT
	TN	SV
	S	AUSGABE
	END	START

Abbildung 9
Programm in der maschinenorientierten Programmiersprache MOPS

3. Die Übersetzung des MOPS-Programms

Für die Umwandlung eines MOPS-Programms in die maschineninterne Darstellung besteht ein Übersetzungsprogramm.
Man bezeichnet das MOPS-Programm auch als Ursprungsprogramm und dessen maschineninterne Darstellung als Objektprogramm.

Durch das Übersetzungsprogramm wird das Ursprungsprogramm in das Objektprogramm umgewandelt.

Das geschieht in vier Phasen:

1. Phase - Eingabe des Ursprungsprogramms
2. Phase - erster Übersetzungslauf
3. Phase - zweiter Übersetzungslauf
4. Phase - Ausgabe des Objektprogramms

Das Übersetzungsprogramm befindet sich auf einem Magnetband. Zur Übersetzung wird es vom Bediener der EDV-Anlage eingelegt und nach einer festgelegten Regieanweisung gestartet. Die vier Phasen gelangen nacheinander in den Hauptspeicher, von wo aus sie abgearbeitet werden.

Ist eine Phase beendet, wird automatisch die nächste vom Magnetband in den Hauptspeicher übertragen und die Abarbeitung ausgelöst.

3.1. Die Eingabe des Ursprungsprogramms

LEKTION 19

Das vom Programmierer aufgestellte Ursprungsprogramm wird Befehl für Befehl in Lochkarten übertragen. Diese Lochkarten werden im ersten Teil der Übersetzung auf ein Magnetband übernommen.

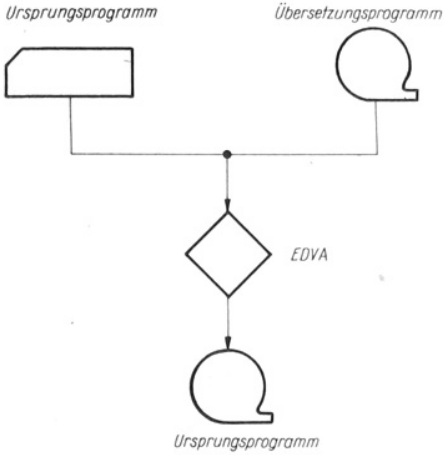


Abbildung 10
Eingabe des Ursprungsprogramms

3.2. Der erste Übersetzungslauf

Das Magnetband mit dem Ursprungsprogramm wird Befehl für Befehl in den Hauptspeicher eingelesen. Es wird überprüft, ob ein Definitions- oder ein Verarbeitungsbefehl vorliegt. Nur diese beiden Befehlsarten sind möglich.

Definitionsbefehle werden ausgeführt, das heißt, im Speicher wird die angegebene Konstante eingetragen, es werden die für einen Bereich erforderlichen Stellen reserviert.

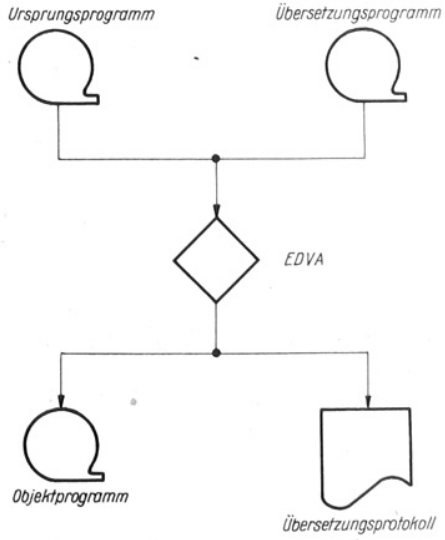


Abbildung 11
Erster Übersetzungslauf

Die Adresse der Konstanten oder des Bereichs ordnet das Übersetzungsprogramm dem im Definitionsbefehl angegebenen Merkmal zu und nimmt beide Angaben in ein Adressenbuch auf.

Im Adressenbuch werden alle im Programm vorkommenden Merkmale mit ihren zugehörigen numerischen Adressen aufgenommen. Sie werden vom Übersetzungsprogramm während des ersten Übersetzungslaufes selbständig festgelegt.

Damit wird das Aufstellen des Speicherbelegungsplans, eine Arbeit die bei Verwendung der Maschinensprache vom Programmierer vorgenommen werden mußte, der EDV-Anlage übertragen.
Für Befehle des Verarbeitungsteils werden die erforderlichen Speicherplätze freigehalten.
Auch für diesen Teil des Programms erfolgt die Speicherplatzzuweisung (sechs Zeichen je Befehl), so daß die Merkmale der Verarbeitungsbefehle mit ihren numerischen Adressen ebenfalls in das Adressenbuch aufgenommen werden können.

3.3. Der zweite Übersetzungslauf

Im zweiten Übersetzungslauf werden die Verarbeitungsbefehle in die maschineninterne Darstellung umgewandelt.
Nachdem das Magnetband mit dem Ursprungsprogramm zurückgespult ist, wird wieder Befehl für Befehl in den Hauptspeicher eingelesen. Die Definitionsbefehle werden übergangen, sie sind bereits im ersten Übersetzungslauf ausgeführt worden.
Zur Umwandlung des Operationsteils existiert im Übersetzungsprogramm eine Liste, in der alle in MOPS zugelassenen Abkürzungen für Operationen mit ihrer zugehörigen numerischen Verschlüsselung enthalten sind. Jede im maschinenorientierten Programm vorkommende Operationsabkürzung wird also durch die maschineninterne Darstellung ersetzt.

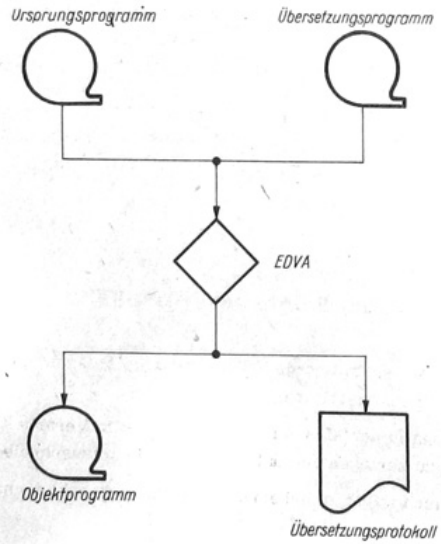


Abbildung 12
Zweiter Übersetzungslauf

Zur Umwandlung der im Adressenteil der Verarbeitungsbefehle angegebenen Merkmale in numerische Adressen wird das im ersten Übersetzungslauf entstandene Adressenbuch herangezogen.
Am Ende des zweiten Übersetzungslaufes ist das Ursprungsprogramm in das Objektprogramm umgewandelt. Zum Erkennen des Programmendes bei der Übersetzung dient der Befehl END.

3.4. Die Ausgabe des Objektprogramms

Um nicht bei jeder Verwendung des Programms den Übersetzungsvorgang wiederholen zu müssen, können wir uns das Objektprogramm wahlweise auf Lochkarten, Lochband oder auf Magnetband ausgeben lassen.
Verwenden wir das Programm wieder, können wir direkt auf das Objektprogramm zurückgreifen und es in die EDV-Anlage eingeben.

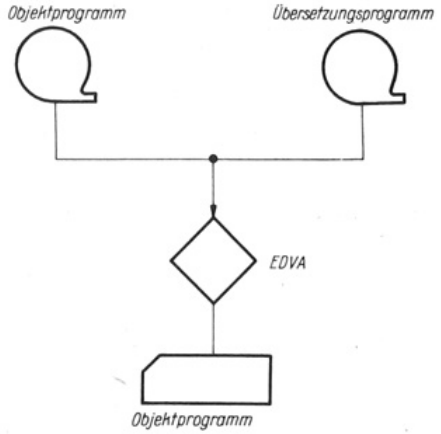


Abbildung 13
Ausgabe des Objektprogramms

Während des ersten und des zweiten Übersetzungslaufes wird auf dem Drukker das Übersetzungsprotokoll ausgegeben. Es enthält in einer Gegenüberstellung die Merkmale und numerischen Adressen (das Adressenbuch), die Befehle in MOPS und in maschineninterner Darstellung sowie die definierten Konstanten und Bereiche.

Das Übersetzungsprotokoll bildet die Verbindung zwischen dem Ursprungsprogramm und dem Objektprogramm.

4. Die Merkmale der Programmierung in maschinenorientierter Programmiersprache

Maschinenorientierte Sprachen enthalten die bei den meisten Anlagen vorhandenen Befehlsarten in allgemeiner Form.

Wie bei der Programmierung in der Maschinensprache sind die Verarbeitungsaufgaben in Einzelschritte zu zerlegen und dafür die Befehle aufzuschreiben.

Die Operationen werden durch leicht zu beherrschende Buchstabenkombinationen angegeben.

LEKTION

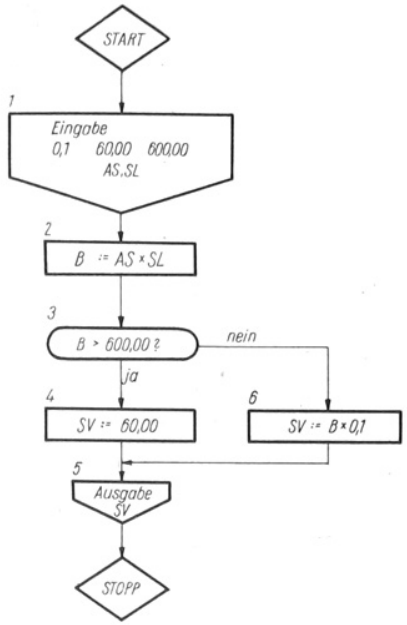
19

Die Angabe von Merkmalen statt numerischer Hauptspeicheradressen ist einfacher und übersichtlicher. Ein Speicherbelegungsplan ist nicht aufzustellen.

Da die EDV-Anlage nur die maschineninterne Darstellung versteht, ist eine Umwandlung des maschinenorientierten Programms erforderlich. Das Übersetzungsprogramm wandelt das Ursprungsprogramm in das Objektprogramm um.
Das Übersetzungsprogramm übernimmt somit einen großen Teil der Arbeiten, die in der Maschinensprache vom Programmierer ausgeführt werden mußten, wie die Speicherbelegung und die interne Verschlüsselung.

Damit haben wir eine wesentliche Erleichterung der Programmierung erreicht.

Da eine EDV-Anlage in der Regel ihre eigenen Befehle hat, läuft auch ein Programm in maschinenorientierter Sprache nur auf allen EDV-Anlagen des gleichen Typs. Daher stammt auch die Bezeichnung maschinenorientierte Sprache.



Daten	Merkmal
0,1	1ZEHNT
60,00	SECHZIG
600,00	SECHS
AS	AS
SL	SL
B	B
SV	SV

Befehl	Symbol
<AC> + <HS>	AD
<AC> · <HS>	M
<AC> > <HS>	VG
<HS> ← <AC>	TV
<AC> ← <HS>	TN
<E> ← <HS>	E
<HS> ← <A>	A
bedingter Sprung (nein)	SN
unbedingter Sprung	S
Halt	H

MERKMAL	OPERATION	ADRESSE

LEKTION

19

Anhang

LEKTION

20

Die Programmaufstellung (III)

- Problemorientierte Programmiersprache -

1. Einleitung

2. Die Programmiersprache ALGOL 60

2. 1. Die Semantik und die Syntax
2. 2. Die Schreibweise arithmetischer Ausdrücke
2. 3. Die Ergibtanweisung
2. 4. Die zusammengesetzte Anweisung
2. 5. Die Vereinbarungen
2. 6. Der Block
2. 7. Die bedingte Anweisung
2. 8. Die Laufanweisung

3. Die Merkmale der Programmierung in problemorientierter Programmiersprache

1. Einleitung

In der vorangegangenen Lektion wurde klar, daß die Programmierung in einer maschinenorientierten Sprache gegenüber der Programmierung in der Maschinensprache zwar eine Vereinfachung mit sich bringt, die Zerlegung der Verarbeitungsaufgabe in Einzelbefehle aber nicht beseitigt.

Die Bemühungen zur Erleichterung der Programmierung führten daher zur Schaffung von problemorientierten Programmiersprachen. Sie stellen gegenwärtig das rationellste Hilfsmittel zur Programmierung von Verarbeitungsaufgaben dar, da ein Zerlegen bis auf Einzelbefehle nicht mehr notwendig ist.

Problemorientierte Sprachen sind maschinenunabhängig. Sie gestatten die Notierung des Programms in einer dem Problem angepaßten Sprache.

LEKTION

20

Sie zeichnen sich durch
- leichte Erlernbarkeit und Handhabung,
- geringen Zeitaufwand für die Programmierung und
- universelle Verwendbarkeit aus.

Unterschiedlich wie die Probleme in Wissenschaft, Technik und Produktion sind auch die problemorientierten Programmiersprachen. Wir erinnern in diesem Zusammenhang an die Datenfernübertragung Leipzig - Moskau - Leipzig anläßlich der Leipziger Frühjahrsmesse 1969. Dabei ging es um die Berechnung der Steuerinformationen für eine numerisch gesteuerte Werkzeugmaschine. Für die Programmierung des dazu notwendigen Ursprungsprogramms wurde eine speziell für die Belange der Werkzeugmaschinen entwickelte Sprache mit dem Namen SYMAP angewendet.
Wir wollen in dieser Lektion am Beispiel der Programmiersprache ALGOL 60 einige Grundprinzipien der problemorientierten Programmiersprachen erläutern und deutlich machen, wie damit die Programmierung vereinfacht werden kann.

2. Die Programmiersprache ALGOL 60

Wie bei den natürlichen Sprachen, so sind auch bei den Programmiersprachen die Wortbedeutung (Semantik) und der Satzbau (Syntax) ausschlaggebend.

2. 1. Die Semantik und die Syntax

Die Semantik legt die Bedeutung der Sprachelemente fest.
Die Syntax bestimmt den Gebrauch der Satzzeichen und die Reihenfolge der Anordnung der Elemente der Sprache.

Eine syntaktische Regel unserer deutschen Sprache besagt zum Beispiel, daß in einer Aufzählung die einzelnen Wörter durch ein Komma zu trennen sind, eine andere, daß ein Fragesatz mit einem Fragezeichen abzuschließen ist.
Ähnliche Regeln gelten auch bei Programmiersprachen. In dieser Lektion gehen wir nur auf die semantischen und syntaktischen Festlegungen der Sprache ALGOL 60 ein, die wir für die Darstellung der Beispiele benötigen. Wir verzichten auch auf eine exakte Definition der verwendeten Begriffe.

Ein Programmierer, der in ALGOL programmieren will, muß die Grundregeln natürlich vollständig beherrschen. Nur wenn er in der Lage ist, die Semantik und die Syntax eindeutig festzulegen, ist gesichert, daß das ALGOL-Programm maschinell in ein solches Maschinenprogramm übersetzt wird, das genau die Maßnahmen durchführt, die der Programmierer beabsichtigt. Jede noch so unerheblich erscheinende Abweichung von den Regeln der Semantik und der Syntax stellt das Übersetzungsprogramm vor eine unlösbare Aufgabe. Es genügt also nicht, daß der Programmierer die Programmiersprache sicher beherrscht, er muß auch peinlich genau und sehr gewissenhaft arbeiten.

LEKTION 20

2.2. Die Schreibweise arithmetischer Ausdrücke

ALGOL 60 wurde vorwiegend für die Programmierung wissenschaftlich-technischer Berechnungen geschaffen. Bei derartigen Aufgaben herrschen mathematische Formeln vor.
Die herkömmlichen Darstellungsarten für arithmetische Ausdrücke sind für eine maschinelle Schreibweise wenig geeignet. Bruchstriche, Exponenten und Indizes bereiten dabei die meisten Schwierigkeiten. Eine Aufgabe bei der Schaffung von ALGOL 60 bestand somit darin, eine Darstellung für arithmetische Ausdrücke zu finden, die eine leichte maschinelle Bearbeitung gestattet.
Wir gehen bei der Programmierung in ALGOL 60 direkt von den arithmetischen Ausdrücken, dem mathematischen Modell der Verarbeitungsaufgabe, aus. Ein Programmablaufplan wird in der Regel für die Aufstellung eines ALGOL-Programms nicht benötigt, lediglich bei sehr umfangreichen Aufgaben wird die Erarbeitung eines Programmablaufplans nicht zu umgehen sein.

Regeln für die ALGOL-Schreibweise arithmetischer Ausdrücke

- Als Bruchstrich wird ein Schrägstrich benutzt.
- Bei Potenzen wird der Exponent hinter einen nach oben gerichteten Pfeil geschrieben.
- Indizes werden hinter die jeweilige Größe in eckige Klammern gesetzt.
- Der Malpunkt wird durch ein liegendes Kreuz dargestellt.
- In arithmetischen Ausdrücken sind nur runde Klammern zulässig.

Beispiel	ALGOL-Schreibweise
$\frac{a}{b}$	a/b
x^3	x↑3
a_i	a[i]
$a \cdot b$	a × b
$[a \cdot (b + c)] \cdot d$	(a × (b + c)) × d
$\frac{(a + b)^2}{(c + d)^3}$	(a + b)↑2/(c + d)↑3
$\frac{1}{2}(x_{i-1} + \frac{a}{x_{i-1}})$	1/2 × (x[i-1] + a / x[i-1])

2.3. Die Ergibtanweisung

Die Ergibtanweisung dient zur Berechnung arithmetischer Ausdrücke, wobei der links vom Ergibtzeichen stehenden Variablen der Wert des rechts davon stehenden arithmetischen Ausdrucks zugewiesen wird.

Bei der Programmaufstellung in ALGOL werden Ergibtanweisungen für die Darstellung arithmetischer Ausdrücke verwendet, in denen Größen berechnet werden, deren Wert nicht von einer Entscheidung abhängt.

Ergibtanweisungen in ALGOL werden mit einem Semikolon abgeschlossen. Für das Ergibtzeichen wird die Schreibweise := benutzt.

LEKTION 20

Regeln für die Reihenfolge der Abarbeitung arithmetischer Ausdrücke

- Gearbeitet wird von links nach rechts unter Beachtung der Klammerregeln:
- Potenzieren,
- Multiplizieren und Dividieren,.
- Addieren und Subtrahieren.

Beispiel

Für die Ergibtanweisung

x:=a↑2×(c+a↑3)/(c-a);

ist der Wert von x zu bestimmen, wenn

a = 2 und c = 7 ist.

Lösung

```
x:=2↑2×(7+2↑3) / (7-2)
x:=4 × (7+8)/5
x:=12
```

2.4. Die zusammengesetzte Anweisung

Sind in einer Aufgabe mehrere verschiedene Größen zu berechnen, die durch eine Folge von Ergibtanweisungen dargestellt werden können, so ordnet man diese hintereinander oder untereinander an. Die syntaktische Regel fordert, die einzelnen Anweisungen durch ein Semikolon zu trennen.
Um auszudrücken, daß die Anweisungen zusammengehören, schließen wir sie in die beiden Worte Beginn und Ende ein. Das ist vor allem für das Übersetzungsprogramm von Bedeutung.

Die Worte Beginn und Ende werden in ALGOL Grundsymbole genannt. Grundsymbole kennzeichnen wir durch Unterstreichen.

Die zusammengesetzte Anweisung hat folgende Struktur:

```
Beginn
        Anweisung;
        Anweisung;
        .
        .
        .
Ende
```

Beispiel

Das in Lektion 16 behandelte Beispiel "Flächen- und Umfangsberechnung eines Rechtecks mit den Seiten a und b" ist in Form einer zusammengesetzten Anweisung darzustellen.

Lösung

```
Beginn
        A:=a×b;
        U:=2×(a+b);
Ende
```

LEKTION 20

2. 5. Die Vereinbarungen

Bei der symbolischen Sprache MOPS mußten wir für die Arbeit des Übersetzungsprogramms jedem Programm einen Vereinbarungsteil beifügen, in dem alle vorkommenden Größen aufzuführen waren. Auch bei ALGOL gehört ein Vereinbarungsteil zum Programm.

Bei ALGOL 60 nimmt der Vereinbarungsteil Typ- und Feldvereinbarungen auf. Er wird mit einem Semikolon abgeschlossen.
Typvereinbarungen legen den Zahlentyp fest.
Mit Hilfe der Feldvereinbarungen wird durch eine Angabe ein Feld – mehrere Speicherplätze – bestimmt.

Bei Typvereinbarungen unterscheiden wir
durch das Grundsymbol ganz - ganze Zahlen,
durch das Grundsymbol reell - reelle Zahlen.

Die jeweiligen Größen sind hinter der Typbezeichnung, durch Komma getrennt, darzustellen. Die letzte Größe wird mit einem Semikolon abgeschlossen.

Die Feldvereinbarung wird
durch das Grundsymbol Feld gekennzeichnet.

Dahinter werden die Bezeichnung des Feldes und in eckigen Klammern, durch Doppelpunkt getrennt, der erste und der letzte Index angegeben.

Stellen die Größen des Feldes ganze Zahlen dar, so ist vor das Grundsymbol Feld noch das Grundsymbol ganz zu setzen.

Beispiel

Für die Darstellung

$a_i,\ i = 1, 2, \ldots 10;$

lautet die Feldvereinbarung

```
Feld a [1:10];
```

2. 6. Der Block

Unter einem Block versteht man eine zusammengesetzte Anweisung mit Vereinbarungsteil.
Der Vereinbarungsteil ist zwischen das Grundsymbol Beginn und die erste Anweisung zu schreiben.

Beispiel

Das Beispiel "Flächen- und Umfangsberechnung eines Rechtecks mit den Seiten a und b" ist als Block zu schreiben.

Lösung

```
Beginn reell a, b, A, U;
       A:=a×b;
       U:=2×(a+b);
Ende
```


2. 7. Die bedingte Anweisung

Eine bedingte Anweisung stellt in ALGOL 60 eine logische Entscheidung mit der danach auszuführenden Anweisung dar.

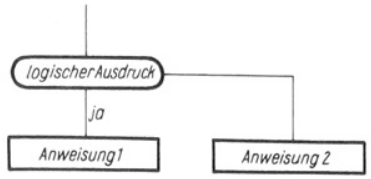


Abbildung 1
Ablaufdarstellung für logische Entscheidung

Für diese Programmablaufdarstellung lautet die bedingte Anweisung:

wenn logischer Ausdruck dann Anweisung 1, sonst Anweisung 2;

Die Wörter wenn, dann und sonst sind Grundsymbole.

Beispiel

Für das Beispiel "Berechnung des Bruttolohns und des SV-Beitrages" ist ein ALGOL-Programm aufzustellen.

Lösung

```
Beginn reell SL, AS, B, SV;
       B:=AS×SL;
       wenn B > 600 dann SV:=60
       sonst SV:=0.1×B;
Ende
```

2. 8. Die Laufanweisung

Bei der Berechnung des Umsatzes der täglich verkauften Waren (vgl. Lektion 17) erhielten wir einen Zyklus mit konstanter Zyklenzahl.
In ALGOL 60 gibt es dafür die Laufanweisung.
Der strukturelle Aufbau der Laufanweisung hat für dieses Beispiel folgende Gestalt:

```
für i:=1 Schrittweite 1 bis n führe aus
U:= U+P[i] × A[i];
```

Die Abarbeitung der Laufanweisung veranlaßt, daß der Reihe nach für $i = 1, 2 \ldots$ bis n der links stehende Wert U berechnet wird.
Der letzte Wert von U liefert die gesuchte Summe des Warenumsatzes, wobei für den Anfangswert U:=0 vorzugeben ist.
Das vollständige ALGOL-Programm für dieses Beispiel lautet:

```
Beginn Feld P[1:n], A[1:n]; reell i, U;
       U:=0;
       für i:=1 Schrittweite 1 bis n führe aus
       U:= U + P[i] × A[i];
Ende
```

Anmerkung:

In der ursprünglichen Definition werden die Grundsymbole von ALGOL 60 in englischer Sprache dargestellt.

LEKTION 20

3. Die Merkmale der Programmierung in problemorientierter Programmiersprache

Problemorientierte Programmiersprachen sind maschinenunabhängig. Sie gestatten eine einfache und rationelle Programmierung, da ein Zerlegen in Einzelbefehle entfällt. Das Programm wird in einer dem zu lösenden Problem angepaßten Sprache aufgestellt. Aus diesem Grunde ist ein Austausch von Programmen auch zwischen Rechenzentren mit unterschiedlichen Typen von EDV-Anlagen möglich.
Wortschatz, Regeln und Vorschriften der problemorientierten Programmiersprache müssen bekannt sein. Sie sind relativ leicht zu erlernen und zu handhaben.
Für die Ausarbeitung von Programmen ist ein Programmablaufplan nur bei umfangreichen Verarbeitungsaufgaben Voraussetzung. In der Regel kann unmittelbar vom mathematischen Modell das Programm abgeleitet werden.

Programme in problemorientierter Programmiersprache müssen mit Hilfe eines Übersetzungsprogramms in die maschineninterne Darstellung übersetzt werden.
Neben den Vorzügen, die zu einer Senkung des Programmieraufwands führen, haben problemorientierte Programmiersprachen aber auch einige Nachteile.
Es sind recht umfangreiche Übersetzungsprogramme erforderlich, um die Ursprungsprogramme in die von der EDV-Anlage verarbeitbare Form der maschineninternen Darstellung zu übersetzen. Auch die daraus entstandenen Programme sind wesentlich umfangreicher als in Maschinensprache geschriebene Programme. Ihr Einsatz ist daher mit längeren Rechenzeiten und einem höheren Speicherplatzbedarf verbunden. Die mit Hilfe von Übersetzungsprogrammen aus einer problemorientierten Programmiersprache übersetzten Programme können in maschineninterner Darstellung zwei- bis dreimal soviel Befehle enthalten, als ein in MOPS erarbeitetes Programm.
Ausschlaggebend ist jedoch, daß durch die Anwendung problemorientierter Sprachen der Mensch von Routinearbeiten weitgehend entlastet und für schöpferische Arbeit, zum Beispiel für das Suchen nach der zweckmäßigsten Lösung, frei wird.
Ein Überblick über die Merkmale der behandelten Programmiersprachen läßt den unterschiedlichen Programmieraufwand deutlich werden.

Merkmale der Programmiersprachen

Merkmale	Maschinen-sprache	Maschinen-orientierte Programmier-sprache	Problem-orientierte Programmier-sprache
Zerlegung in Einzelbefehle notwendig	ja	ja	nein
Speicherplan notwendig	ja	nein	nein
Angabe der Adresse im Befehl durch	Nummer der Speicherzelle	Namen	entfällt
Numerierung der Befehle notwendig	ja	nein	nein
Angabe der Sprungadresse durch	Nummer der Speicherzelle	Marken	Marken
Übersetzungsprogramm notwendig	nein	ja	ja

LEKTION 20

LEKTION

21

Die Praxis der Programmierung

1. Einleitung

2. Die Methoden der praktischen Arbeit der Programmierung

2. 1. Die Programmzerlegung
2. 2. Die Programmtestung
2. 2. 1. Der Trockentest
2. 2. 2. Der Programmtest an der EDV-Anlage
2. 3. Die Auswahl der Programmiersprache

3. Die Programmdokumentation

3. 1. Die Aufgaben der Programmdokumentation
3. 2. Der Inhalt der Programmdokumentation
3. 2. 1. Das Karteiblatt
3. 2. 2. Die ausführliche Dokumentation
3. 3. Die Aufbewahrung der Datenträger und der Kartei

4. Die Programmbibliothek

4. 1. Die Bedeutung der Programmbibliothek
4. 2. Die Organisation der Programmbibliothek

5. Die "Benutzergemeinschaft elektronische Datenverarbeitung"

1. Einleitung

In den vergangenen Lektionen haben wir die verschiedenen Möglichkeiten der Programmierung kennengelernt. Die hier gewonnenen Erkenntnisse wollen wir nun auf die praktische Arbeit des Programmierers anwenden.

Die Programmierung bildet eine Etappe der Lösung eines Problems mit Hilfe elektronischer Datenverarbeitungsanlagen.
In jeder Etappe der Einsatzvorbereitung hat der Mensch geistig-schöpferische Arbeit zu leisten.
Der Organisator bearbeitet ein Problem bis zum Programmablaufplan.

Der Programmierer fertigt nach dem vom Organisator erarbeiteten Programmablaufplan das Programm an.
Deshalb müssen Programmierer und Organisator eng zusammenarbeiten.

Das vom Programmierer aufgestellte Programm bildet die Grundlage für die unmittelbare Arbeit an der Anlage. Folglich muß auch zu den Bedienkräften der EDV-Anlage ein enger Kontakt bestehen. Sie testen das Programm. Stellen sich dabei Fehler heraus, geht das Programm mit entsprechenden Hinweisen an den Programmierer zurück, der die Korrektur vornimmt.
In dieser Lektion beschäftigen wir uns mit der Praxis der Programmierung und einigen damit im Zusammenhang stehenden Problemen, wie der Programmbibliothek und der "Benutzergemeinschaft elektronische Datenverarbeitung".

LEKTION

21

2. Die Methoden der praktischen Arbeit der Programmierung

2. 1. Die Programmzerlegung

In den Beispielen der vergangenen Lektionen haben wir zum besseren Verständnis recht einfache Programme erläutert. Die in der Praxis zu lösenden Planungsaufgaben, Optimierungsprobleme oder Netzplanaufgaben sind jedoch sehr umfangreich. Ein Programmierer würde zur Ausarbeitung eines solchen Programms viele Monate, vielleicht sogar Jahre benötigen. In der Zwischenzeit geht die Entwicklung weiter, viele Bedingungen können sich ändern. Die Folge wäre eine Überarbeitung des Programms, die erneut Zeit erfordert. Eine derartige Arbeitsweise wäre ein ernstes Hindernis für eine effektive Nutzung der elektronischen Datenverarbeitung in unserer Volkswirtschaft. Es ist daher zweckmäßig und notwendig, ein Programm weitestgehend in Teile zu zerlegen. So können die einzelnen Teile von mehreren Programmierern nebeneinander bearbeitet werden. Dadurch verkürzt sich die Dauer der Programmaufstellung wesentlich.
Durch die Programmzerlegung wird es auch möglich, Unterprogramme herauszufinden, die bereits vorliegen und auf die zurückgegriffen werden kann, so daß eine Programmierung für diese Teile entfällt.
Teilprogramme haben außerdem den Vorteil, daß sie leichter zu überblikken sind. Es empfiehlt sich, jedes Teilprogramm sofort zu testen. Ein Trockentest hilft beim Erkennen versteckter Programmierfehler.
Die Programmzerlegung setzt natürlich voraus, daß ein erfahrener Programmierer die Arbeiten koordiniert und anleitet.

Bei umfangreichen Datenverarbeitungsprojekten ist die Programmzerlegung unumgänglich.

2. 2. Die Programmtestung

Die Erarbeitung eines Programms birgt Fehlerquellen in sich. Deshalb ist jedes Programm zu erproben, um sicher zu sein, daß es keine Fehler enthält.

Das Aufspüren und Beseitigen von Fehlern in einem Programm bezeichnet man als Programmtestung.

Die Programmtestung geschieht auf verschiedene Weise.

2. 2. 1. Der Trockentest

Aus den vorangegangenen Lektionen ist bekannt, daß wir bei einem Trockentest mit Papier und Bleistift ein Programm auf seine Richtigkeit überprüfen.
Je nach Notwendigkeit entscheidet der für die Verarbeitungsaufgabe verantwortliche Programmierer, ob außer dem Trockentest für die einzelnen Programmteile noch ein Trockentest des gesamten Programms durchzuführen ist.

2. 2. 2. Der Programmtest an der EDV-Anlage

Der Test an der Anlage erfolgt in der Regel in mehreren Durchläufen. Da Testläufe wertvolle Maschinenzeit erfordern – eine Stunde Laufzeit der EDV-Anlage kostet bis zu 2 000, - Mark – müssen sie gut vorbereitet werden.

Die Prüfinformationen

Schon bei der Niederschrift des Programms werden an geeigneten Stellen Prüfinformationen vorgesehen. Bei der Testung des Programms an der Anlage ermöglichen diese Prüfinformationen eine Kontrolle der Zwischenergebnisse und der Zyklen.
Die Prüfinformationen müssen eindeutig und ausreichend sein. Deshalb ist es unumgänglich, auch Text mit auszugeben, um den Programmablauf schnell und sicher rekonstruieren zu können. Text sollte zur Kennzeichnung numerischer Zwischenergebnisse und der durchlaufenen Programmteile verwendet werden.
Zur Ausgabe der Prüfinformationen dient der Schnelldrucker.

Die Testbeispiele

Von Bedeutung für die Erprobung eines Programms sind die Testbeispiele. Sie dürfen einerseits nicht zu kompliziert und umfangreich sein, damit sie überschaubar bleiben, andererseits ist zu fordern, daß alle, wenigstens aber alle wesentlichen Zweige des Programms durchlaufen werden.

Die Prüfprogramme

Bei den Testläufen kommt es darauf an, die Maschinenzeit möglichst effektiv zu nutzen. Deshalb gibt es für viele Anlagen spezielle Prüfprogramme. Beispielsweise werden syntaktische Fehler in Programmiersprachen von solchen Prüfprogrammen angezeigt. In ALGOL ist ein häufig vorkommender syntaktischer Fehler, daß vergessen wird, benutzte Größen zu vereinbaren.

Protokollprogramme drucken Zwischenergebnisse und Zieladressen ausgeführter Sprünge aus, so daß an Hand dieser Angaben der Programmablauf rekonstruiert werden kann. Mit einem anderen Prüfprogramm kann der Inhalt von Speicherbereichen des Hauptspeichers ausgegeben werden. Der Ausdruck der Prüfprogramme erfolgt über Schnelldrucker.

2. 3. Die Auswahl der Programmiersprache

In den Lektionen 18-20 behandelten wir die Merkmale der verschiedenen Programmiersprachen. In der Praxis der Programmierung ist zu entscheiden, welche Programmiersprache im Einzelfall zweckmäßig ist.
Vom Standpunkt der Schnelligkeit der Verarbeitung sind der Maschinencode und die maschinenorientierte Programmiersprache gleichwertig, da in beiden Fällen die Operationsteile der Maschinenbefehle verwendet werden und jeder Befehl in maschinenorientierter Sprache einem Befehl im Maschinencode entspricht.
Weil aber die Programmierung im Maschinencode langwierig und zeitraubend ist, sollte der Maschinencode nur für das Übersetzungsprogramm verwendet werden.
Alle übrigen oft verwendeten Programme können in maschinenorientierten Sprachen, beispielsweise in MOPS beim Robotron 300, programmiert werden. Da die Adressen nur als Bezeichnungen anzugeben sind, braucht sich der Programmierer nicht um die Einzelheiten der Speicherzuordnung zu kümmern, wodurch das Programmieren und insbesondere das Ändern von Programmen wesentlich erleichtert wird.
Am einfachsten für die Programmierung sind problemorientierte Sprachen, beispielsweise ALGOL. Die in diesen Sprachen geschriebenen Programme laufen zwar langsamer ab als die in symbolischer Sprache formulierten, aber infolge der Schnelligkeit moderner EDV-Anlagen fällt dieser Umstand meist nicht allzusehr ins Gewicht.

Regeln für die Auswahl der Programmiersprache

1. Soll ein Programm oft verwendet werden oder
   ist das Programm für maximale Rechengeschwindigkeit auszulegen,
   weil die Resultate sehr schnell zur Verfügung stehen müssen,
   muß eine maschinenorientierte Programmiersprache verwendet werden.
   In allen übrigen Fällen wird in einer problemorientierten Sprache programmiert.
2. Der Maschinencode wird nur für Übersetzungsprogramme benutzt. Diese werden aber mit der Anlage mitgeliefert, beispielsweise das MOPS-Übersetzungsprogramm des Robotron 300.

3. Die Programmdokumentation

Die Programmdokumentation gibt eine detaillierte Übersicht über alle mit der Bearbeitung eines Programms auf einer EDV-Anlage zusammenhängenden Fragen, angefangen von der Problemstellung bis hin zu den Testbeispielen.

3. 1. Die Aufgaben der Programmdokumentation

Die Programmdokumentation hat im wesentlichen drei Aufgaben zu erfüllen:

1. Mit der Programmdokumentation wird den Bedienkräften der EDV-Anlage eine klare und ausführliche Arbeitsanweisung für die Bearbeitung des Programms gegeben.

2. Eine einwandfreie Programmdokumentation ist Voraussetzung für notwendig werdende Programmänderungen. Ein Programm wird meist nicht nur für eine einmalige Verwendung erarbeitet, es soll für die gleiche Aufgabenstellung immer wieder genutzt werden. Änderungen an bereits getesteten Programmen sollten zwar so selten wie möglich vorgenommen werden, sie sind aber manchmal nicht zu umgehen. Im Verlauf der Arbeit kann sich herausstellen, daß mit wenig Aufwand das Programm wesentlich verbessert werden kann. Gesetzliche Vorschriften können sich ändern, zum Beispiel die Gewinnberechnung, die dann eine Änderung aller einschlägigen Programme nach sich ziehen.
3. Die Programmdokumentation ermöglicht sowohl den Programmaustausch zwischen Rechenzentren, die über gleiche Anlagen verfügen, als auch den Austausch von Programmen in problemorientierten Sprachen. Ohne eine einwandfreie Programmdokumentation wäre ein Programmaustausch nur schwer zu verwirklichen.

3. 2. Der Inhalt der Programmdokumentation

Alle zur Programmdokumentation gehörenden Unterlagen werden zweckmäßigerweise in einer Programmappe zusammengefaßt. Sie besteht aus einem Karteiblatt für die Kurzinformation, der ausführlichen Dokumentation sowie aus der Kartei, die Aufschluß darüber gibt, wo die zum Programm gehörenden Datenträger zu finden sind.

3. 2. 1. Das Karteiblatt

Von jedem Programm wird ein Karteiblatt angelegt, das eine kurze Übersicht über das Programm gibt. Es dient zugleich als Titelblatt für die Programmappe. Auf dem Karteiblatt sind anzugeben:

1. Programmbezeichnung
2. Name des Programmierers
3. Benötigte Maschinenausrüstung
4. Speicherbedarf
5. Verwendete Wahlschalter
6. Rechenzeit
7. Kurzbeschreibung

Die Kenntnis des Speicherbedarfs und der Art der verwendeten Wahlschalter des Programms ist wichtig, wenn mehrere Programme – gewissermaßen als Bausteine – zu einem großen Programm zusammengefügt werden.

3. 2. 2. Die ausführliche Dokumentation

Sie umfaßt:

1. Angaben über Problemstellung und Größe der lösbaren Probleme
2. Angaben über den Aufbau der Datenträger
3. Programmbeschreibung
4. Programm (in der Regel als Schnelldruckerausschrift)
5. Regieanweisungen für die Bedienkraft an der EDV-Anlage
6. Testbeispiele mit Ergebnissen der Testrechnungen einschließlich der von der EDV-Anlage gelieferten Rechenprotokolle
7. Programmänderungen (Art und Zeitpunkt der Änderung, Name des Programmierers, der die Änderung durchführte)

Die Programmbeschreibung

Sie umfaßt:

1. Aufgabe
2. Lösungsmethode
3. Zahlendarstellung
4. Genauigkeit
5. Verwendete Standardprogramme
6. Aufbau der Datenträger
7. Rechenzeit
8. Bemerkungen

3. 3. Die Aufbewahrung der Datenträger und der Kartei

Beim Robotron 300 existieren die Programme als Lochkartensätze. Sie werden in Stahlschränken in Karteikästen aufbewahrt. Die einzelnen Kästen enthalten Programme ähnlicher Aufgabenstellung, sie sind entsprechend der Einordnung beschriftet. Innerhalb der Kästen kennzeichnen Karteireiter die einzelnen Programme.
Häufig werden Programme auf Magnetbändern aufbewahrt. Jede Magnetbandspule wird mit einer Nummer versehen. Aus der entsprechenden Karteikarte ist zu ersehen, welches Programm sich auf dem Magnetband befindet und ab welchem Zeitpunkt es mit anderen Daten beschrieben werden kann.

Diese Kartei wird natürlich nicht nur für die Magnetbänder geführt, die zu einer Programmdokumentation gehören, sondern auch für alle anderen Magnetbänder, um ein zeitraubendes Suchen nach freien Magnetbändern zu vermeiden.

Magnetbandkartei	Band-Nr.: 295	
Belegt mit	Verfallsdatum	Unterschrift
Testdaten Kap.-bilanz 1969	30. 9. 69 frei	
Programm Kap.-bilanz (Progr. Nr. 3. 6. 01. 01. 2)	30. 10. 69 frei	
Programm Quadratwurzel (Progr. Nr. 3. 3. 05. 03. 0)	31. 6. 70	

LEKTION 21

4. Die Programmbibliothek

Unter einer Programmbibliothek verstehen wir eine geordnete Sammlung erprobter Programme von allgemeiner Verwendbarkeit einschließlich der Programmdokumentation, um die Nutzung eines Programms durch einen anderen als den Verfasser zu ermöglichen.

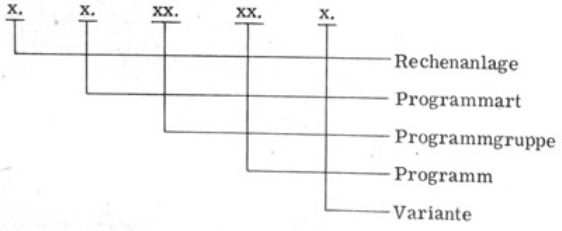
4.1. Die Bedeutung der Programmbibliothek

Programmbibliotheken entstanden aus dem Bestreben heraus, einmal erarbeitete Programme recht oft wieder zu benutzen. Wir konnten uns in den vorangegangenen Lektionen davon überzeugen, wie zeitaufwendig Programmierarbeiten sind. Die Kosten für die Anfertigung von umfangreichen Programmen können im Laufe der Zeit die Anschaffungskosten der Datenverarbeitungsanlage erreichen und sogar übertreffen. Aus diesem Grunde ist es notwendig, einmal vorhandene und erprobte Programme für die gleiche Problemstellung wieder zu verwenden. Das setzt voraus, daß eine einwandfreie Dokumentation vorliegt; sie bildet die Grundlage für eine Information über das Programm.

4.2. Die Organisation der Programmbibliothek

Die Programmbibliothek unterscheidet sich formal nicht von anderen Bibliotheken. Auch hier herrscht eine feste Ordnung, die sich auf die spezifische Art des Inhalts bezieht.
Anfangs bestanden die Programmbibliotheken aus vielen Einzelprogrammen, die sich oft sehr ähnlich waren. Je komplexer aber die Datenverarbeitungsanlagen werden, je mehr Systeme angeboten werden, desto komplexer werden auch die Programmbibliotheken.
Die Programme sind nach Sachgebieten geordnet. Jedes Programm erhält eine Nummer. Sie erscheint auf allen zum Programm gehörenden Unterlagen (Programmdokumentation).

Aufbau der Programmnummer



5. Die „Benutzergemeinschaft elektronische Datenverarbeitung"

Aus dem Bestreben heraus, die einzelnen Programme, die in den Programmbibliotheken der Betriebe und Institute aufbewahrt werden, möglichst vielen Anwendern zugängig zu machen und Doppelarbeiten zu vermeiden, entstand die "Benutzergemeinschaft elektronische Datenverarbeitung".

In der Deutschen Demokratischen Republik gehören ihr alle Institutionen an, die elektronische Datenverarbeitungsanlagen besitzen. Ihre Zusammenarbeit gliedert sich in Benutzergruppen entsprechend den Anlagentypen. So besteht auch eine Benutzergruppe der EDV-Anlagen Robotron 300. In ihnen arbeiten die einzelnen Institutionen kameradschaftlich zusammen und unterstützen sich gegenseitig.
Die Zusammenarbeit in den Benutzergruppen dient

- dem Erfahrungsaustausch über Probleme der Einsatzvorbereitung,
- dem Austausch von Projekten und Programmen sowie
- der Information über geplante Programme und Projekte, um Doppelarbeiten auszuschalten.

Die Zusammenarbeit erstreckt sich auch auf Erfahrungsaustausch und gegenseitige Hilfe auf technischem Gebiet, beispielsweise bei der Wartung der Datenverarbeitungsanlagen sowie bei Havariefällen.
Zur Benutzergemeinschaft gehört eine Projekt- und Programmzentrale. Sie nimmt die Aufgaben der Programmbibliothek wahr, indem sie alle mit einer elektronischen Datenverarbeitungsanlage bereitzustellenden Programme sammelt, aufbewahrt und an den einzelnen Anwender ausliefert. Darüber hinaus werden von der Projekt- und Programmzentrale alle von den Anwendern elektronischer Datenverarbeitungsanlagen erarbeiteten Projekte und Programme dokumentarisch erfaßt und zur breiten Nachnutzung vermittelt. Zu diesem Zweck sind alle Mitglieder der Benutzergemeinschaft verpflichtet, geplante und abgeschlossene Projekte und Programme in einer festgelegten Form der Projekt- und Programmzentrale zu melden.
An allen Beteiligten liegt es, die Arbeit der Benutzergemeinschaft so zu entwickeln, daß für den einzelnen Anwender und für die gesamte Volkswirtschaft die höchste Effektivität beim Einsatz der elektronischen Datenverarbeitung erzielt wird, eine der vorrangigen Aufgaben der wissenschaftlich-technischen Revolution und des entwickelten gesellschaftlichen Systems des Sozialismus.

LEKTION 21

Einige Fragen

zur Vorbereitung auf die Prüfung zum Fernsehkurs "Elektronische Datenverarbeitung"

Wiederholungsfragen zum Themenkomplex 4
"Grundlagen der Programmierung"

1. Was versteht man unter dem Begriff "Problemanalyse"?
   Wozu ist sie notwendig?
2. Erklären Sie den Begriff "Algorithmus"!
   Geben Sie ein Beispiel für einen Algorithmus!
3. Wozu dient ein Programmablaufplan?
4. Nennen Sie die wesentlichen Sinnbilder zur Aufstellung eines Programmablaufplans!
5. Unterscheiden Sie
   - unverzweigte Geradeausprogramme,
   - verzweigte Geradeausprogramme,
   - zyklische Programme!
6. Was verstehen Sie unter dem Trockentest eines Programmablaufplans?
7. Nennen Sie wesentliche Befehle einer EDV-Anlage!
8. Erläutern Sie die Merkmale der Programmierung in einer Maschinensprache und in einer maschinenorientierten Sprache!
9. Warum muß ein Programm in maschinenorientierter Sprache zur Abarbeitung auf der EDV-Anlage in die Maschinensprache umgewandelt werden?
   Wie erfolgt diese Umwandlung?
10. Nennen Sie Vor- und Nachteile der problemorientierten Programmiersprache!
11. Erläutern Sie Inhalt und Bedeutung der Programmdokumentation!
12. Was versteht man unter einer Programmbibliothek und worin besteht ihre Bedeutung?

Wissenswertes

für die Teilnehmer des Fernsehkurses "Elektronische Datenverarbeitung"

Seit dem vergangenen Jahr erscheinen zur Zeitschrift "Rechentechnik/Datenverarbeitung" Beihefte, in denen bestimmte Themen im Komplex behandelt werden. Dieses Material hat sich unter anderem bei Vorbereitungskollektiven zum Einsatz von Datenverarbeitungsanlagen oder bei der Qualifizierung auf dem Gebiet der elektronischen Datenverarbeitung gut bewährt. Folgende Hefte wurden herausgegeben:

Erfahrungen aus der Einsatzvorbereitung für die EDVA Robotron 300 aus dem VEB Berliner Glühlampenwerk (1/1968)

Erfahrungen, die das Zentrum für Organisation und Rechentechnik des Betriebes als Erstanwender einer EDVA Robotron 300 in 3jähriger Arbeit gesammelt hat, werden hier mit dem Ziel vermittelt, künftigen Nutzern von Datenverarbeitungsanlagen bei den Vorbereitungs- und Organisationsarbeiten Hilfe zu geben.

Maschinenorientierte Systemunterlagen für Robotron 300 (2/1968)

Hier wird ein Teil der maschinenorientierten Programmbibliothek für den Robotron 300 beschrieben, womit der Anwender Informationen über die Leistungsfähigkeit und Konzeption einzelner Programme erhält.

Erfahrungen von Erstanwendern mit EDVA (1/1969)

Betrieben und Institutionen wird Anleitung gegeben, wie die Einsatzvorbereitung rationalisiert und effektiv gestaltet werden kann. Dabei werden Fragen wie Qualifizierung, ökonomische Stimuli, strukturelle Entwicklung der Einsatzkollektive, methodische Probleme, Datenerfassung, Programmablaufplanung und Programmierung, Änderungsdienst, Planung und Leitung der Umstellung sowie Ökonomie der Datenverarbeitung behandelt.

Einsatz der EDVA für die Information und Dokumentation (2/1969)

Dieses Heft behandelt die Einsatzvorbereitung einer EDVA Robotron 300 für Information und Dokumentation; Mikrofilm, Datenverarbeitung und Dokumentation; Faktenrecherchesystem für die Prognose; Datenerfassung; Datenverarbeitungsgerechte Systematik zum Zwecke der Information und Dokumentation und andere Themen.

Die Beihefte zur "Rechentechnik/Datenverarbeitung" – Preis je Heft 5,- M bei einem Umfang von 64 Seiten – können vom Verlag Die Wirtschaft, 1055 Berlin, Am Friedrichshain 22, Abteilung Vertrieb, bezogen werden. Am sichersten aber ist ein Abonnement bei der Deutschen Post.

Elektronische Datenverarbeitung

Was sollte der sozialistische Leiter
von der EDV wissen?

Von Prof. Dr. rer. oec. Heinz Steuer

Etwa 112 Seiten · 23 grafische Darstellungen
Broschur · etwa 3,00 M

Der Einsatz moderner Instrumente der Leitung, wie der elektronischen Datenverarbeitung, ist ein objektives Erfordernis des ökonomischen Systems des Sozialismus in der DDR. Datenverarbeitungssysteme müssen deshalb so konzipiert werden, daß sie ein aktives Element der Leitungs- und Organisationssysteme werden und auf den Erkenntnissen der marxistisch-leninistischen Organisationswissenschaft aufbauen.

Diese Zusammenhänge erläutert Prof. Dr. Steuer in populärer Weise. Er setzt keine Fachkenntnisse der EDV voraus, sondern ordnet die elektronische Datenverarbeitung in das Gebiet der sozialistischen Wirtschaftsführung ein, er arbeitet Probleme heraus, die der Leiter in seiner zukunftsbezogenen und strategischen Arbeit kennen und beherrschen muß.

Mehr als 20 Schemata erleichtern die Verständlichkeit des Stoffes.

VERLAG
DIE WIRTSCHAFT
BERLIN

Dozent kommt ins Haus

Fernsehen macht das Lernen leichter: Es bringt den Dozenten des EDV-Kurses in Ihre Wohnung.
Das Fernsehen vermittelt Ihnen aber nicht nur berufliche Qualifizierung, es bietet Ihnen noch viel mehr. Das wissen Sie sicherlich.
Wissen Sie aber auch, daß Sie beim Kauf eines Fernsehgerätes günstige Teilzahlungsbedingungen in Anspruch nehmen können?

Wir beraten Sie gern.

VEB INDUSTRIEVERTRIEB RUNDFUNK UND FERNSEHEN MIT DEM NETZ SEINER FACHFILIALEN

RFT RADIO-*television*

Programmablaufplanung

Fachkunde für Datenverarbeiter

Etwa 150 Seiten · 130 Abbildung. · Broschur · etwa 4,75 M

In dieser Broschüre wird eine Vielzahl von Beispielen aus Gebieten wie Statistik, Ökonomie, Mathematik so aneinandergereiht, daß ihnen ein Algorithmus zugrunde liegt. Zuerst werden einfache Beispiele mit Detailproblemen dargestellt. Dann werden die Einzelprobleme in weiteren Beispielen nach unterschiedlichen Gesichtspunkten miteinander kombiniert. Auf diese Art und Weise wird ständig der Grad der Komplexität erhöht. Das methodische Vorgehen nach der Broschüre zeigt dem Lernenden, wie man grundsätzlich bei der Lösung von komplexen Aufgabenstellungen zur Programmablaufplanung vorzugehen hat. So kann der der Broschüre zugrunde liegende "Lehralgorithmus" zum "Lösungsalgorithmus" für den Lernenden werden.

VERLAG
DIE WIRTSCHAFT
BERLIN